Social-Enhanced Attentive Group Recommendation

Da Cao, Xiangnan He, Lianhai Miao, Guangyi Xiao*, Hao Chen, Jiao Xu

Abstract—With the proliferation of social networks, group activities have become an essential ingredient of our daily life. A growing number of users share their group activities online and invite their friends to join in. This imposes the need of an in-depth study on the group recommendation task, i.e., recommending items to a group of users. Despite its value and significance, group recommendation remains an unsolved problem due to 1) the weights of group members are crucial to the recommendation performance but are rarely learnt from data; 2) social followee information is beneficial to understand users' preferences but is rarely considered; 3) user-item interactions are helpful to reinforce the performance of group recommendation but are seldom investigated.

Toward this end, we devise neural network-based solutions by utilizing the recent developments of attention network and neural collaborative filtering (NCF). First of all, we adopt an attention network to form the representation of a group by aggregating the group members' embeddings, which allows the attention weights of group members to be dynamically learnt from data. Secondly, the social followee information is incorporated via another attention network to enhance the representation of individual user, which is helpful to capture users' personal preferences. Thirdly, considering that many online group systems also have abundant interactions of individual users on items, we further integrate the modeling of user-item interactions into our method. Through this way, the recommendation for groups and users can be mutually reinforced. Extensive experiments on the scope of both macro-level performance comparison and micro-level analyses justify the effectiveness and rationality of our proposed approaches.

Index Terms—Group Recommendation, Attention Network, Social Followee Information, Neural Collaborative Filtering.

1 Introduction

Recommender systems have played a pivotal role in the development of network technology owing to their outstanding ability in mitigating the information overload problem. Both consumers and service providers benefit from the development of recommender systems, which can improve user experience and help service providers to adjust their strategy to create new business opportunities. Existing studies on recommender systems are mainly focused on recommending items to individual users, arousing several research topics, such as context-aware recommendation [1], graph structure [2], [3], [4], [5], and domain-specific applications [6], [7]. However, traditional recommender systems are specifically designed for optimizing user-item interactions and are not optimized for some complex circumstances such as recommending items to a group of users, denoted as *group recommendation*. Group activities are very popular in various real-world scenarios, such as a group of travellers join in a a travel plan on Mafengwo[1], a group of teenagers participate in a social party on Meetup[2], and a group of researchers discuss a paper on Mendelay[3].

Group decision making is a dynamic and interactive process among individuals, and each member of a group could contribute to the final decision results. Therefore, group recommendation should not only consider individuals' preferences, but also take the group decision process into account. Generally, the preferences of a group are obtained by aggregating the preferences of its members via a predefined aggregation strategy, such as average [8], least misery [9], maximum satisfaction [10], and expertise [11]. However, we argue that these predefined strategies are data independent, lacking the flexibility to dynamically adjust the weights of group members. This flexibility is particularly useful when a group makes decision on items of different types. As such, these aggregation strategies are insufficient to capture the complicated and dynamic process of group decision making, resulting in the suboptimal performance for group recommendation. If a group recommender system suggests items without considering the dynamic and interactive process among group members, it may end up with unsuitable items and adversely hurt groups' experience.

Despite its value and significance, group recommendation remains in its infancy due to the following challenges: 1) Existing approaches handling group recommendation largely applied a predefined and fixed strategy to aggregate the preferences of group members, which is insufficient to capture the complicated and dynamic process of group decision making. Therefore, how to devise an adaptive

*Guangyi Xiao is the Corresponding Author.

- *D. Cao, L. Miao, G. Xiao and H. Chen are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China 410082. E-mail: caoda0721@gmail.com, lianhaimiao@gmail.com, guangyi.xiao@gmail.com, chenhao@hnu.edu.cn*
- *X. He is with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China 230026. E-mail: xiangnanhe@gmail.com*
- *J. Xu is with the Central Research Institute, CVTE Inc., No.6, the 4th Yunpu Road, Guangzhou, Guangdong, China 510530. Email: xujiao@cvte.com*

1. https://www.mafengwo.com
2. https://www.meetup.com
3. https://www.mendeley.com

strategy to dynamically endow weights for group members is a non-trivial task. 2) Many group-aware social platforms also have abundant data of social followee relations, an important data source to reflect user personal preferences. Thus how to leverage social followee relations to improve user (i.e., group member) representation is of great interest. 3) A group's preferences over an item also manifests its members' preferences over the item, and vice versa. Therefore, how to devise an unified framework to reinforce the recommendation performance of both group-item and user-item is a valuable research issue. In summary, to serve both groups and users with a high-quality recommendation service, it is highly desirable to develop techniques that can comprehensively consider the factors of adaptive weight learning, social followee influence, and jointly recommending items for groups and users.

To address these challeges, we devise neural network-based solutions to investigate the group recommendation issue comprehensively. First of all, we employ the recent advance in neural network modeling — the attention mechanism [12], [13] — to enable group members contribute differently to the group decision. The attention weights for group members are dynamically adjusted when the group interacts with different items. Afterwards, the social followee information is further incorporated into the user representation learning via another attention network. The group-level and user-level attention networks are connected with a hierarchical structure, which is helpful for reinforcing the representation learning of both groups and users. Furthermore, both group-item and user-item interactions are embedded into the neural collaborative filtering (NCF) framework [14], such that the recommendation performance of group-item and user-item can be mutually enhanced. By conducting experiments on two real-world datasets, we demonstrate that our proposed framework yields significant gains as compared with state-of-the-art competitors.

A preliminary version of this work has been published as a conference paper in SIGIR 2018 [15]. This paper is significantly different from its preliminary version in the methodology. Specifically, this work approaches the group recommendation via jointly considering the influence of group member and social followee, but our previous work [15] only focuses on the impact of group member. Moreover, we design a hierarchical attention network to learn the representations of groups, which is an extension of previous single-layer attention network. As such, the Related Work (Section 2), the Methods (Section 3), and the Experiments (Section 4) have been re-written to support our solution to the new generic problem.

The main contributions of this work are summarized as follows:

- We explore the promising yet challenging problem of group recommendation. To the best of our knowledge, this is the first group recommender system that leverages neural attention network to learn the aggregation strategy from data in a dynamic way.
- The representation learning for individual users is further enhanced by utilizing their social followee information. Moreover, user-item interactions are further integrated. The group-item and user-item recommendation performance can be mutually reinforced.
- Extensive experiments are performed on two real-world datasets to demonstrate the effectiveness of our methods. Meanwhile, the datasets and codes are released to facilitate the research community[4].

2 Related Work

2.1 Group Recommendation

Group recommendation has received a lot of attentions in recent years and has been widely applied in various domains. Technically speaking, these work can be divided into two categories — *memory-based* and *model-based* approaches.

Memory-based approaches can be further subdivided into *preferences aggregation* [16] and *score aggregation* [17]. Preferences aggregation strategy first aggregates the profiles of group members into a new profile, and then employs recommendation techniques designed for individuals to make group recommendation. Score aggregation strategy first predicts the individuals' scores over candidate items, and then aggregates the predicted scores of members in a group via predefined strategies (e.g., average, least misery, maximum satisfaction and so on) to represent the group's preferences. However, both approaches are predefined and inflexible, which utilize trivial methods to aggregate members' preferences.

Distinct from memory-based approaches, model-based methods exploit the interactions among members by modeling the generative process of a group. The PIT model [18] effectively identifies the group preference profile for a given group by considering the personal preferences and personal impacts of group members. A probabilistic model named COM [17] is proposed to model the generative process of group activities and make group recommendations. A deep learning-based algorithm AGR [19] is presented to learn the influence weight of each user in a group to perform the group recommendation. Hu et al. [20] proposed a deep learning approach DLGR , which aims to learn high-level comprehensive features to represent group preference so as to avoid the vulnerabilities in a shallow representation. Yin et al. [21] proposed a social influence-based group recommendation algorithm SIGR , which is able to learn both user embedding and user social influences from data in a unified way. Moreover, some other sophisticated essentials have be investigated for the group recommendation, such as multimedia content [22], probabilistic method [23], [24], and user behavior [11]. Our work falls into the model-based category. In addition to learn the high-level interactions among users, groups, and items under the deep learning framework, our work employs the attention mechanism as the underlying principle for the aggregation of users' embedding representations. Meanwhile, user-item and group-item recommendations are mutually reinforced under our framework.

2.2 Social-Enhanced Recommender Systems

As the rich information on social network becomes available, social-enhanced recommender systems have draw extensive attention in the research community, which aim

4. https://github.com/caoda0721/SoAGREE

to improve the recommendation performance by exploiting the social influence. Moreover, social-enhanced recommender systems are capable of handling various application scenarios by considering some sophisticated factors, such as privacy [25], tag information [26], and user preference [27].

In fact, there exists a variety of literature that attempts to apply the social influence to the group recommendation scenario. A social-aware group recommendation framework is presented in [22] that jointly utilizes social relationships and social behaviors to not only infer a group's preference, but also model the tolerance and altruism characteristics of group members. For event recommendation, a Bayesian latent factor model is proposed in [28] that considers social group influence and individual preference simultaneously. The work of [29] introduces a preference-oriented social networks to capture the correlation of preference rankings between individuals who interact in social networks. A trust induced recommendation mechanism is investigated in [30], which generates personalized advices for the inconsistent experts to reach higher consensus in group decision making. In this article, to model the influence of social followee information, we regard social followees as attributes for each user and employ the attention mechanism as the attribute aggregation strategy, which is significantly different from previous social-enhanced recommender systems.

2.3 Deep Learning for Recommendation

The proliferation of deep learning has swept the research community, among which recommender systems are no exception. The majority of work that integrates recommender systems with deep learning methods primarily utilized deep neural networks for modeling auxiliary information. The features learnt by deep neural networks are then incorporated into collaborative filtering algorithms. Different from previous work, there are some attempts that try to seamlessly combine recommender systems with deep learning methods by modeling user-item interactions [14] and higher-order interactions among features [31]. The success of NCF [14] has been further extended to attribute-based social recommendation [32], being utilized as the foundation of our work as well.

The attention mechanism with the realization of neural networks has been shown effective in several tasks, such as image processing [33] and question answering [34]. It simulates human recognition by focusing on some selective parts of the whole image or the whole sentence while ignoring some other informative (less informative) parts. In fact, the attention mechanism has been investigated in the field of recommender systems. To get the representation for a multimedia item (e.g., image or micro-video [35]), Chen et al. [12] aggregate its components (e.g, regions or frames) with an attention network. Then the similar attention mechanism is applied to aggregate interacted items to get user representation to make recommendation. Attentive collaborative filtering [12] introduced the item- and component-level attention model for multimedia recommendation. In attentional factorization machines [13], the weights of feature interactions are learnt via neural attention network. Meanwhile, the work of [36] proposed a neural attentive item similarity model for item-based recommendation, which is capable of distinguishing which historical items in a user profile are more important for a prediction via the attention network. Moreover, Sun et al. [37] present an attentive recurrent network-based approach for temporal social recommendation [38], which models users' complex dynamic and general static preferences over time by fusing social influence among users with two attention networks. Inspired by these pioneering work, the key idea of our framework is to regard a group as an image or a sentence and learn to assign attention weights for members (components) in the group (image or sentence): higher weights indicate that the corresponding members (components) are significant to the end task (image or sentence).

3 METHODS

Generally speaking, our proposed solution consists of two components: 1) hierarchical attention network learning which utilizes dual-level attention network to represent groups and users in a hierarchical structure; and 2) interaction learning with NCF which recommends items for both users and groups. For clarity, we employ the attention mechanism as the group member aggregation strategy to perform the group recommendation and ignore the social influence in this stage, and denote it as AGREE (short for "**A**ttentive **G**roup **RE**comm**E**ndation"). Thereafter, we merge the social influence into the group recommendation scenario by utilizing another attention network to aggregate the followees of each user, and denote it as SoAGREE (short for "**So**cial-enhanced **A**ttentive **G**roup **RE**comm**E**ndation").

3.1 Notations and Problem Formulation

We propose to address the group recommendation problem under the representation learning framework [39]. Under the representation learning paradigm, each entity of interest is represented as an embedding vector, which encodes the inherent properties of the entity (e.g., semantics of a word, interests of a user etc.) and is to be learned from data.

Suppose we have n_u users $\mathcal{U} = \{u_1, u_2, ..., u_{n_u}\}$, n_g groups $\mathcal{G} = \{g_1, g_2, ..., g_{n_g}\}$, n_f followee users $\mathcal{F} = \{f_1, f_2, ..., f_{n_f}\}$, and n_i items $\mathcal{V} = \{v_1, v_2, ...v_{n_i}\}$. The l-th group $g_l \in \mathcal{G}$ is consisted of a set of users, i.e., group members with user indexes $\mathcal{K}_l = \{k_{l,1}, k_{l,2}, ..., k_{l,|g_l|}\}$, where $u_{k_{l,*}} \in \mathcal{U}$, and $|g_l|$ is the size of the group. The i-th user $u_i \in \mathcal{U}$ follows a set of followees with followee indexes $\mathcal{H}_i = \{h_{i,1}, h_{i,2}, ..., h_{i,|u_i|}\}$, where $f_{h_{i,*}} \in \mathcal{F}$, and $|u_i|$ is the number of followees. There are four kinds of observed interaction data among $\mathcal{U}, \mathcal{G}, \mathcal{F}$, and $\mathcal{V}$, namely, user-item interactions, group-item interactions, group-user interactions, and user-followee interactions. Then, given a target group (or target user), our task is defined as recommending a list of items that the group (or the user) may be interested in.

3.2 Hierarchical Attention Network Learning

The first component of our SoAGREE framework is a hierarchical attention network that models the aggregations with respect to the user in *group-level* and the followee in *user-level*. We use two attention sub-networks to learn these aggregations jointly. Figure 1 illustrates the structure of our proposed hierarchical attention network. Given the

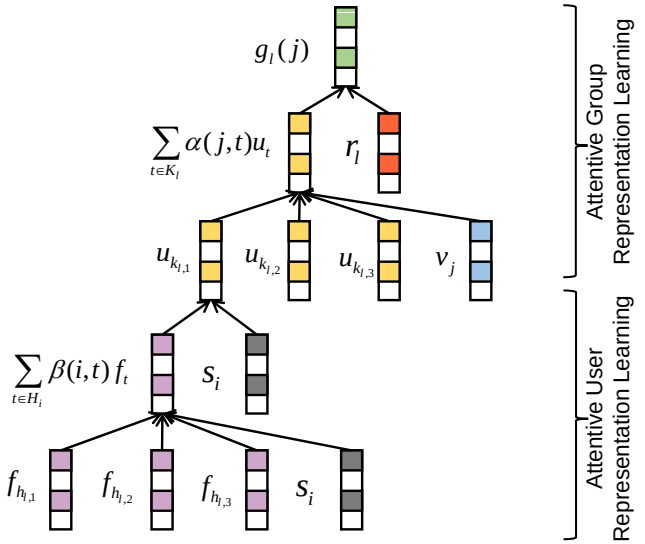


Fig. 1: Illustration of the hierarchical attention network learning, which is composed of the attentive group representation learning and the attentive user representation learning.

t-th member in the l-th group, the t-th followee for the i-th user, and the j-th item, we use $\alpha(j,t)$ to denote the t-th member's preference degree on item j, and $\beta(i,t)$ to denote the t-th followee's impact degree on user i.

3.2.1 Attentive Group Representation Learning

The group-level attention is employed to select group members that are representative to the group, and then the representations of informative group members are aggregated to characterize the group. Let $\mathbf{u}_i$ and $\mathbf{v}_j$ be the embedding vector for user u_i and item v_j, respectively, which are basic representation blocks in our AGREE model. Our target is to obtain an embedding vector for each group to estimate its preference on an item. To learn dynamic aggregation strategy from data, it is necessary to define the group embedding as dependent of the embeddings of its member users and the target items, which can be abstracted as,

$$\mathbf{g}_l(j) = f_g(\mathbf{v}_j, \{\mathbf{u}_t\}_{t\in\mathcal{K}_l}), \tag{1}$$

where $\mathbf{g}_l(j)$ denotes the embedding of group g_l tailored for predicting its preference on target item v_j, $\mathcal{K}_l$ contains the user indexes of group g_l, and f_g is the aggregation function to be specified. In AGREE, we design the group embedding as consisting of two components — user embedding aggregation and group preference embedding:

$$\mathbf{g}_l(j) = \underbrace{\sum_{t\in\mathcal{K}_l}\alpha(j,t)\mathbf{u}_t}_{\text{user embedding aggregation}} + \underbrace{\mathbf{r}_l}_{\text{group preference embedding}}. \tag{2}$$

Next we elaborate the two components.

We perform a weighted sum on the embeddings of group g_l's member users, where the coefficient $\alpha(j,t)$ is a learnable parameter denoting the influence of member user u_t in deciding the group's choice on item v_j. Intuitively, if a user has more expertise on an item (or items of the similar type), she should have a larger influence on the group's choice on the item [11]. To understand this, let us consider an example that a group discusses which city to travel to; if a user has traveled to China many times, she should be more influential when the group considers whether should travel to a city in China. Since in the representation learning framework, embedding $\mathbf{u}_t$ encodes the member user's historical preference and embedding $\mathbf{v}_j$ encodes the target item's property, we parameterize $\alpha(j,t)$ as a neural attention network with $\mathbf{u}_t$ and $\mathbf{v}_j$ as the input:

$$\begin{aligned} a(j,t) &= \mathbf{h}^T\text{ReLU}(\mathbf{P}_v\mathbf{v}_j + \mathbf{P}_u\mathbf{u}_t + \mathbf{b}), \\ \alpha(j,t) &= \text{softmax}(a(j,t)) = \frac{\exp a(j,t)}{\sum_{t'\in\mathcal{K}_l}\exp a(j,t')}, \end{aligned} \tag{3}$$

where $\mathbf{P}_v$ and $\mathbf{P}_u$ are weight matrices of the attention network that convert item embedding and user embedding to hidden layer, respectively, and $\mathbf{b}$ is the bias vector of the hidden layer. We use ReLU as the activation function of the hidden layer, and then project it to a score $a(j,t)$ with a weight vector $\mathbf{h}$. Lastly, we normalize the scores with a softmax function, which is a common practice in neural attention network [12], [13], [40]; it makes the attention network a probabilistic interpretation, which can also deal with groups of different sizes in our case. With such a soft attention mechanism, we allow each member user to contribute in a group's decision, where the contribution of a user is dependent on her historical preference and the target item's property, which are learned from past data of group-item interactions and user-item interactions (to be discussed in Section 3.3).

Besides aggregating the embeddings of group members, we further associate a group g_l with a dedicated embedding $\mathbf{r}_l$. The intention is to take the general preference of a group into account. Our consideration is that in some cases when users form a group, they may pursue a target that is different from the preference of each user. For example, in a family of three, the child prefers cartoon movie and the parents favor romantic movie; but when they go to a cinema together, the final chosen movie could be an educational movie. As such, it is beneficial to associate a group with an embedding to denote its general preference, in addition to the one aggregated from its members. To combine the components of group preference embedding with user embedding aggregation, we perform a simple addition operation, same as the previous work [12], [41] that combine different signals in the embedding space. Our empirical results in Section 4.5 show that this component can significantly improve the group recommendation performance.

3.2.2 Attentive User Representation Learning

Social followee information is incorporated into the attentive user representation learning component, in which the followees of a user are regarded as her attributes and are aggregated with attention weights. Let $\mathbf{f}_i$ be the embedding vector for followee f_i, which is the basic ingredient for inferring the social impact. The target of the attentive user representation learning component is to obtain an embedding vector for each individual user. We define the user embedding as the function with the input of its followees' embeddings, which is formally defined as,

$$\mathbf{u}_i = f_u(\{\mathbf{f}_t\}_{t\in\mathcal{H}_i}), \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2936475, IEEE Transactions on Knowledge and Data Engineering

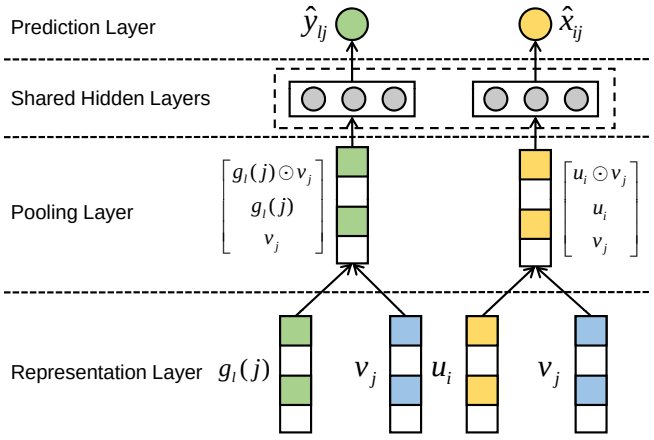


Fig. 2: Illustration of the NCF-based interaction learning, which follows a joint optimization scheme.

where $\mathbf{u}_i$ denotes the embedding of user u_i, $\mathcal{H}_i$ contains the followee indexes of user u_i, and f_u is the aggregation function to be specified. In SoAGREE, we define the user embedding as the combination of followee embedding aggregation and user preference embedding:

$$\mathbf{u}_i = \underbrace{\sum_{t \in \mathcal{H}_i} \beta(i,t)\mathbf{f}_t}_{\text{followee embedding aggregation}} + \underbrace{\mathbf{s}_i}_{\text{user preference embedding}}. \quad (5)$$

Similar to the group-level attention, the user-level attention score for the t-th followee of user u_i is also a two-layer network and is formularized as:

$$b(i,t) = \mathbf{h}^T \text{ReLU}(\mathbf{Q}_f \mathbf{f}_t + \mathbf{Q}_s \mathbf{s}_i + \mathbf{b}),$$
$$\beta(i,t) = \text{softmax}(b(i,t)) = \frac{\exp b(j,t)}{\sum_{t' \in \mathcal{H}_i} \exp b(j,t')}, \quad (6)$$

where $\mathbf{Q}_f$ and $\mathbf{Q}_s$ are weight matrices of the attention network that convert followee embedding and user preference embedding to hidden layer, respectively, and $\mathbf{b}$ is the bias vector of the hidden layer. ReLU is utilized as the activation function of the hidden layer, and is then projected to a score $\beta(i,t)$ with a weight vector $\mathbf{h}$. Finally, the user-level attention is normalized with a softmax function. With the underlying principle of attention mechanism, the attention weights of a user's followees are dynamically learnt, and the user's followees contribute unequally in forming the representation for the user. In addition, the user embedding is associated with a user preference embedding $\mathbf{s}_i$, which takes the general preference of a user into account. Through this way, the followee embedding aggregation and user preference embedding are mutually reinforced. Experimental results in Section 4.5 show that this design can significantly enhance the group recommendation performance.

3.3 Interaction Learning with NCF

NCF is a multi-layer neural network framework for item recommendation [14]. Its idea is to feed user embedding and item embedding into a dedicated neural network (which needs to be customized) to learn the interaction function from data. As neural networks have strong ability to fit the data, the NCF framework is more generalizable than the traditional MF model, which simply applies a data-independent inner product function as the interaction function. As such, we opt for the NCF framework to perform an end-to-end learning on both embeddings (that represent users, items, and groups) and interaction functions (that predict user-item and group-item interactions).

Figure 2 illustrates our customized NCF solution. Since we aim to achieve both recommendation tasks for groups and users simultaneously, we design the solution to learn the user-item and group-item interaction functions together. Specifically, given a user-item pair (u_i, v_j) or a group-item pair (g_l, v_j), the representation layer first returns the embedding vector for each given entity (details see Section 3.2). Then the embeddings are fed into a pooling layer and hidden layers (shared by the two tasks) to obtain the prediction score. Next we elaborate the two components.

3.3.1 Pooling layer

Assuming the input is a group-item pair (g_l, v_j), the pooling layer first performs element-wise product on their embeddings, i.e., $\mathbf{g}_l(j)$ and $\mathbf{v}_j$, and then concatenates it with the original embeddings:

$$\mathbf{e}_0 = \varphi_{pooling}(\mathbf{g}_l(j), \mathbf{v}_j) = \begin{bmatrix} \mathbf{g}_l(j) \odot \mathbf{v}_j \\ \mathbf{g}_l(j) \\ \mathbf{v}_j \end{bmatrix} \quad (7)$$

The rationale is twofold. 1) The element-wise product subsumes MF, which uses multiplication on each embedding dimension to model the interation between two embedding vectors; moreover, element-wise product has been demonstrated to be highly effective in feature interaction modeling in low-level of neural architecture [31]. 2) Nevertheless, the element-wise product may lose some information in the original embeddings which may be useful for later interaction learning. To avoid such information loss, we further concatenate it with the original embeddings.

Note that such a pooling operation is partially inspired from the state-of-the-art neural recommender model NeuMF [14], which shows that combines MF with MLP in the hidden layer leads to better performance. As MLP concatenates the original user embedding and item embedding, it inspires us to keep the original embeddings to facilitate the learning of later hidden layers. We apply the same pooling operation for the input of a (u_i, v_j) pair.

3.3.2 Shared Hidden layers

Above the pooling layer is a stack of fully connected layers, which enable the model to capture the nonlinear and higher-order correlations among users, groups, and items.

$$\begin{cases} \mathbf{e}_1 = \text{ReLU}(\mathbf{W}_1 \mathbf{e}_0 + \mathbf{b}_1) \\ \mathbf{e}_2 = \text{ReLU}(\mathbf{W}_2 \mathbf{e}_1 + \mathbf{b}_2) \\ \ldots\ldots \\ \mathbf{e}_h = \text{ReLU}(\mathbf{W}_h \mathbf{e}_{h-1} + \mathbf{b}_h) \end{cases}, \quad (8)$$

where $\mathbf{W}_h$, $\mathbf{b}_h$, and $\mathbf{e}_h$ denote the weight matrix, bias vector, and output neurons of the h-th hidden layer, respectively. We use the ReLU function as the non-linear activation function, which has empirically shown to work well. Moreover, we use the tower structure for hidden layers and leave

the further tuning on the structure as future work. Finally, the output of the last hidden layer $\mathbf{e}_h$ is transformed to a prediction score via:

$$\begin{cases} \hat{x}_{ij} = \mathbf{w}^T \mathbf{e}_h, \ if \ \mathbf{e}_0 = \varphi_{pooling}(\mathbf{u}_i, \mathbf{v}_j) \\ \hat{y}_{lj} = \mathbf{w}^T \mathbf{e}_h, \ if \ \mathbf{e}_0 = \varphi_{pooling}(\mathbf{g}_l(j), \mathbf{v}_j) \end{cases}, \tag{9}$$

where $\mathbf{w}$ denotes the weights of the prediction layer; $\hat{x}_{ij}$ and $\hat{y}_{lj}$ represent the prediction for a user-item pair (u_i, v_j) and a group-item pair (g_l, v_j), respectively.

It is worth mentioning that we have purposefully designed the prediction of the two tasks share the same hidden layers. This is because that the group embedding is aggregated from user embeddings, which makes them in the same semantic space by nature. Moreover, this can augment the training of group-item interaction function with user-item interaction data and vice versa, which facilitates the two tasks reinforcing each other.

3.4 Model Optimization

3.4.1 Objective Function

Since we address recommendation task from the ranking perspective, we opt for pairwise learning method for optimizing model parameters. The assumption of pairwise learning is that an observed interaction should be predicted with a higher score than its unobserved counterparts. Specifically, we employ the regression-based pairwise loss, which is a common choice in item recommendation [32]:

$$\mathcal{L}_{user} = \sum_{(i,j,s)\in\mathcal{O}} (x_{ijs} - \hat{x}_{ijs})^2 = \sum_{(i,j,s)\in\mathcal{O}} (\hat{x}_{ij} - \hat{x}_{is} - 1)^2, \tag{10}$$

where O denotes the training set, in which each instance is a triplet (i, j, s) meaning that user u_i has interacted with item v_j, but has not interacted with item v_s before (i.e., v_s is a negative instance sampled from the unobserved interactions of u_i); $\hat{x}_{ijs} = \hat{x}_{ij} - \hat{x}_{is}$, means the margin of the prediction of observed interaction (u_i, v_j) and unobserved interaction (u_i, v_s). Since we focus on implicit feedback, where each observed interaction has a value of 1 and unobserved interaction has a value of 0, we have $x_{ijs} = x_{ij} - x_{is} = 1$.

We are aware that another prevalent pairwise learning method in recommendation is the Bayesian Personalized Ranking (BPR) [42], [43]. It is worth pointing out that an advantage of the above regression-based pairwise loss over BPR is that it eliminates the need of tuning the L_2 regularization for the weights in the hidden layers (i.e., $\{\mathbf{W}_h\}$ and $\mathbf{w}$). In BPR, the loss for an instance (i, j, s) is formulated as $-\log \sigma(\hat{x}_{ij} - \hat{x}_{is})$, where σ is the sigmoid function. To decrease the BPR loss on a multi-layer model, a trivial solution is to scale up the weights in each update. As such, it is crucial to enforce the L_2 regularization on the weights to avoid this trivial solution. In contrast, our chosen loss optimizes the margin term $\hat{x}_{ij} - \hat{x}_{is}$ towards 1, making such a trivial solution fail to decrease the loss. Thus the weights can be learned without any constraint on it.

Similarly, we can obtain the pairwise loss function for optimizing the group recommendation task:

$$\mathcal{L}_{group} = \sum_{(l,j,s)\in\mathcal{O}'} (y_{ljs} - \hat{y}_{ljs})^2 = \sum_{(l,j,s)\in\mathcal{O}'} (\hat{y}_{lj} - \hat{y}_{ls} - 1)^2, \tag{11}$$

where O' denotes the training set for the group recommendation task, in which each instance (l, j, s) means that group g_l has interacted with item v_j, but has not interacted with v_s before.

3.4.2 Learning Details

We present some learning details that are important to replicate our method.

Mini-batch training. We perform mini-batch training, where each mini-batch contains both user-item and group-item interactions. Specifically, we first shuffle all observed interactions, and then sample a mini-batch of observed interactions. For each observed interaction, we sample a fixed number of negative instances to form the training instances.

Pre-training. It is known that neural networks are rather sensitive to initialization [14]. To better train AGREE and SoAGREE, we pre-train them with a simplified version that removes the attention networks, i.e., assigning uniform weights on user embeddings and followee embeddings to obtain the group embedding and user embedding, respectively. With the pre-trained model as an initialization, we further train the AGREE and SoAGREE model. Note that we employ Adam [44] in the pre-training phase, which has a fast convergence owing to its adaptive learning rate strategy. After pre-training, we use the vanilla SGD, a common choice in fine-tuning a pre-trained model.

Dropout. As the neurons of fully connected layers can easily co-adapt [31], [45], we employ dropout to improve our solution's generalization performance. Specifically, in the pooling layer, we randomly drop ρ percent of the $\mathbf{e}_0$ vector. Moreover, we also apply dropout on the hidden layer of the neural attention network and the hidden layers of NCF interaction learning component. Note that dropout is only used during training (i.e., computing gradients with back-propagation), and must be disabled during the prediction phase.

4 Experiments

In this section, we conduct extensive experiments on one self-collected dataset and one public dataset to answer the following five research questions:

- **RQ1** How is the effectiveness of our designed attention networks? Can they provide better group recommendation performance?
- **RQ2** How does our proposed AGREE and SoAGREE approaches perform as compared with state-of-the-art group recommender systems?
- **RQ3** How do different predefined settings (e.g., the number of negative samples and dropout ratio) affect our framework?
- **RQ4** How do the two components of group representation (i,e., user embedding aggregation and group preference embedding) and two components of user representation (i.e., followee embedding aggregation and user preference embedding) contribute to the performance of our solutions?

4.1 Experimental Settings

4.1.1 Datasets

We experimented with two real-world datasets, one is crawled from a tourism website Mafengwo[5] and the other one is a publicly accessible dataset released from the competition of context-aware movie recommendation[6].

1. Mafengwo. Mafengwo is a tourism website where users can record their traveled venues, create or join a group travel. We retained the groups which have at least 2 members and have traveled at least 3 venues, and collected their traveled venues. The traveled venues of each group member were also collected. Based on the above criteria, we obtained $5,275$ users, 995 groups, $1,513$ items, $39,761$ user-item interactions, and $3,595$ group-item interactions. The user-item interaction matrix has a sparsity of 99.50%, while the group-item interaction matrix has a sparsity of 99.76%. On average, each group has 7.19 users, each user has traveled 7.54 venues, and each group has traveled 3.61 venues.

Moreover, we crawled the followee information from the Mafengwo platform. For the aforementioned $5,275$ users, we collected their followees. Ultimately, we obtained $5,275$ users, $13,096$ followees, and $53,235$ user-followee interactions. On average, each user has 10.09 followees, and each followee has been followed by 4.06 users.

2. CAMRa2011. CAMRa2011 is a real-world dataset containing the movie rating records of individual users and households. Since the majority of users have no group information in the dataset, we filtered them out and retained users who have joined a group. The user-item interactions and group-item interactions are explicit feedback with the rating scale of 0 to 100. We transformed the rating records to positive instances with the target value of 1 and left the other missing data as negative instances with the target value of 0. The final dataset contained 602 users, 290 groups, $7,710$ items, $116,344$ user-item interactions, and $145,068$ group-item interactions. The user-item interaction matrix has a sparsity of 97.49%, while the group-item interaction matrix has a sparsity of 93.51%. On average, each group has 2.08 users, each user has watched 193.26 movies, and each group has watched 500.23 movies.

As both datasets only contain positive instances (i.e., observed interactions), we randomly sampled from missing data as negative instances to pair with each positive instance. Previous efforts have shown that increasing the negative sampling ratio from 1 to larger values is beneficial to the Top-N recommendation [14]. For AGREE and SoAGREE on both datasets, the optimal sampling ratio is around 4 to 6, so we fixed the negative sampling ratio as 4. Specifically, for each log of Mafengwo, we randomly sampled 4 venues that the user (group) has never visited; for each log of CAMRa2011, we randomly sampled 4 movies that the user (group) has never watched. Each negative instance is assigned to a target value of 0. It is well worth to mention that the social followee information only exists on the Mafengwo dataset. Therefore, the performance of SoAGREE is only evaluated on Mafengwo.

5. http://www.mafengwo.cn
6. http://2011.camrachallenge.com/2011

4.1.2 Evaluation Protocols

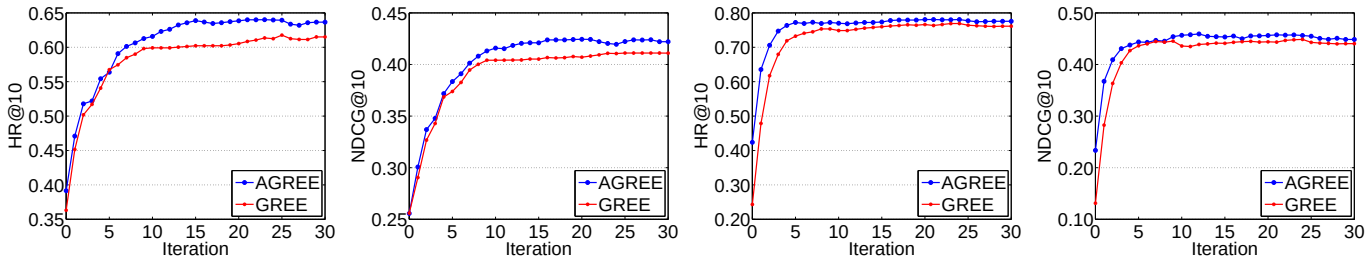
We adopted the *leave-one-out* evaluation protocol, which has been widely utilized to evaluate the performance of the Top-N recommendation [12], [42]. Specifically, for each user (group), we randomly removed one of her (its) interactions for testing. This results in disjoint training set $\mathcal{S}_{train}$ and testing set $\mathcal{S}_{test}$. Since it is too time-consuming to rank all items for each user and group, we followed the common scheme [14] that randomly selected 100 items that were not interacted by the user or the group and ranked the testing item among the 100 items. To evaluate the performance of the Top-N recommendation, we employed the widely used metric — Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). Large values indicate better performance. In leave-one-out evaluation, HR measures whether the testing item is ranked in the Top-N list (1 for yes and 0 for no), while NDCG accounts for the position of the hit by assigning higher score to hit at top positions.

4.1.3 Baselines

To justify the effectiveness of our methods, we compared them with the following methods.

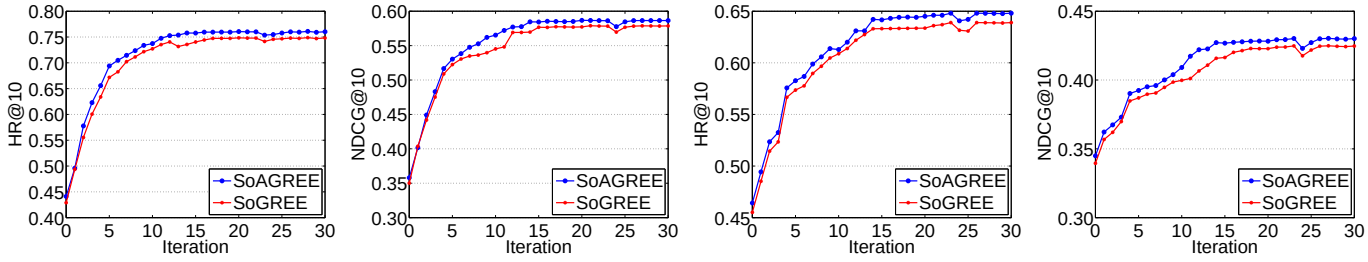
- **NCF.** This method treats a group as a virtual user and ignores the member information of the group [46]. Then users and virtual users are embedded into our NCF solution with the same hyper-parameter setting of AGREE.
- **Popularity [47].** This method recommends items to users and groups based on the popularity of items. The popularity of an item is measured by its number of interactions in the training set. It is a non-personalized method to benchmark the performance of other personalized methods.
- **COM [17].** This is a group-oriented recommender system, which is based on the probability theory to model the generative process of group activities. We used the implementation released by the authors and modified the evaluation codes to adapt our testing scenario. We fine-tuned the hyper-parameters to obtain the optimal result.
- **UL_All [16].** This is a group recommendation algorithm, which involves proposing an upward leveling aggregation method to consider deviations for group recommendations. This method is re-implemented and the testing scheme is modified to be consistent with our setting.
- **AGR [19].** This is an attention-based group recommendation solution. It learns the attention weight of a user by considering the impact of other group members, which ignores the influence of items. A pairwise ranking loss is employed to optimize the solution. We re-implemented this approach and modified its evaluation strategy.
- **GREE and SoGREE.** These are variants of our AGREE and SoAGREE methods. GREE removes the attention network in AGREE, while SoGREE removes the user-level attention network in SoAGREE. Uniform weights are employed. This is to demonstrate the effect of learning varying weights for group members and social followees.
- **AGREE-S.** This is a variant of our AGREE framework by utilizing separate hidden layers to infer user-item interactions and group-item interactions. This method is employed to illustrate the effect of shared hidden layers in interaction learning.
- **AGREE-G.** This is a variant of our AGREE solution by using group member information to enhance the represen-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2936475, IEEE Transactions on Knowledge and Data Engineering



(a) Mafengwo — HR@10 **(b) Mafengwo — NDCG@10** **(c) CAMRa2011 — HR@10** **(d) CAMRa2011 — NDCG@10**

Fig. 3: Performance of AGREE and GREE in each training iteration on both Mafengwo and CAMRa2011 datasets for group-item recommendation (Section 4.2).



(a) User — HR@10 **(b) User — NDCG@10** **(c) Group — HR@10** **(d) Group — NDCG@10**

Fig. 4: Performance of SoAGREE and SoGREE in each training iteration on the Mafengwo dataset for both user-item recommendation and group-item recommendation (Section 4.2).

tation learning of users. Instead of utilizing the followees of a user, this method employs the users who have appeared in a same group with the user to construct the representation for her. This is to manifest the effect of group information in constructing user embeddings.

It should be noted that we were aware of PIT [18], DLGR [20], and SIGR [21], state-of-the-art solutions for group recommendation. However, the authors did not release their implementation and the algorithms are difficult to re-implement. Therefore, these methods are not compared at this stage.

To justify the usefulness of learning the aggregation strategy from data, we further compare with another line of methods that apply a predefined score aggregation strategy. For these methods, we first run the NCF method with the same hyper-parameter setting of AGREE to predict the individual preference scores, and then apply the aggregation strategy to get the group preference score.

- **NCF+avg [8].** NCF+avg is short for "NCF combined with average". It is the simplest aggregation strategy that averages the preference scores of individuals as the group preference score. The hypothesis behind this method is that each member contributes equally to the final group decision.
- **NCF+lm [9].** NCF+lm applies the least misery strategy. It tries to please all members in a group, which uses the minimum score of individuals as the group preference score. The underlying assumption is that the least satisfied member determines the final group decision, which is similar to the well-known *cask principle*.
- **NCF+ms [10].** NCF+ms employs the maximum satisfaction strategy. In contrast to NCF+lm, it tries to maximize the satisfaction of group members. It averages the individual scores above a specified threshold as the group preference score. In this work, we assumed a member prefers to follow other members' options, and treated the maximum score as the preference of the group.
- **NCF+exp [11].** NCF+exp adopts the expertise scheme. It applies a weighted average on individual scores, where the weight reflects the expertise of the user. In our experiments, the expertise of a user is defined as the number of items she has interacted with in the training set.

4.1.4 *Implementation and Hyper-Parameter Setting*

We implemented our method based on PyTorch[7]. For hyper-parameter tuning, we randomly sampled one interaction for each user and group as the validation set. As have mentioned before, the negative sampling ratio was set to 4. For the initialization of the embedding layer, we applied the Glorot initialization strategy [48], which was found to have a good performance. For hidden layers, we randomly initialized their parameters with a Gaussian distribution of a mean of 0 and a standard deviation of 0.1. We used the Adam optimizer for all gradient-based methods, where the mini-batch size and learning rate were searched in $[128, 256, 512, 1024]$ and $[0.001, 0.005, 0.01, 0.05, 0.1]$, respectively. In neural attention network and NCF, we empirically set the size of the first hidden layer same as the embedding size with the dimension of 32, and employed three layers of a tower structure and ReLU activation function. We repeated each setting for 5 times and reported the average results. We further conducted the paired two-sample t-test on NDCG based on the 5 times experiment results.

4.2 Effect of Attention (RQ1)

The primary motivation of this work is to learn variable attention weights for group members and user followees, rather than the commonly used uniform weighting strategy.

7. http://www.pytorch.org

TABLE 1: Case studies of a sampled group on the effect of group-level attention (Section 4.2). The member weights and prediction scores of the group for positive venues (Venue #30, #32, #106) and negative venues (Venue #65, #121, #123) are shown (Section 4.2).

	Model	User #805	User #806	User #807	$\hat{y}$
Venue #30	GREE	0.333	0.333	0.333	0.260
	AGREE	0.286	0.302	**0.412**	**0.572**
Venue #32	GREE	0.333	0.333	0.333	0.096
	AGREE	0.222	**0.583**	0.195	**0.370**
Venue #106	GREE	0.333	0.333	0.333	0.192
	AGREE	**0.364**	0.287	0.347	**0.318**
Venue #65	GREE	0.333	0.333	0.333	0.132
	AGREE	**0.408**	0.311	0.281	**0.091**
Venue #121	GREE	0.333	0.333	0.333	0.132
	AGREE	0.335	**0.374**	0.291	**0.053**
Venue #123	GREE	0.333	0.333	0.333	0.109
	AGREE	0.288	**0.411**	0.301	**0.063**

Therefore, in order to investigate the effectiveness of the attention network, we compare the performance of AGREE with the GREE baseline and SoAGREE with the SoGREE baseline.

Figure 3 shows the performance of AGREE and GREE in each training iteration under the optimal parameter settings. We have the following observations: 1) Compared with GREE, AGREE achieves a relative improvement on both datasets with respect to both metrics. The improvements are statistically significant and mainly stem from the strong representation power of the attention network. 2) Both AGREE and GREE converge rather fast, reaching their stable performance around the 20th iteration. Compared with GREE, AGREE additionally uses an attention network to re-weight the embedding vectors of group members. This improves generalization without affecting the convergence speed, which provides evidence on the effectiveness and rationality of AGREE. Similar experimental results of SoAGREE and SoGREE are revealed in Figure 4. SoAGREE betas SoGREE by a great margin for both user-item recommendation and group-item recommendation.

Micro-Level Analysis. Apart from the superior recommendation performance, another key advantage of AGREE is its ability in interpreting the attention weights of group members. To demonstrate this, we performed some micro-level case studies. To be specifically, we implemented AGREE (SoAGREE) in a two-stage scheme. After obtaining the GREE (SoGREE) model, we fixed the parameters and trained the attention network only to make the effect of attention more distinct. Prediction scores of the group (user) toward positive and negative items are investigated.

We randomly selected a testing group which consists of three users (#805, #806, and #807), and the group has traveled three venues (#30 Argentina, #32 Chile, and #106 Bolivia) with the target value of 1. Each group member also has her owning traveling history[8]. Besides traveled venues, we also randomly picked three negative venues (#65 Iran, #121 Qiandao Lake, and #123 Baoji) with the target value of 0. Table 1 shows the attention weights and prediction score for the group of GREE and AGREE. We have the following observations: 1) For different target venues, the attention weights of group members vary significantly in AGREE. For example, when predicting the group's preference on negative venues #121 and #123, the attention weights of user #806 are relatively high. This is probably because that the user has traveled a lot of Chinese venues (#547 Jiangxi, #62 Yunnan, and #553 Hunan), and thus she has more power in deciding whether the group should travel to other Chinese venues (note that #121 Qiandao Lake and #123 Baoji are Chinese venues). 2) For positive venues, the prediction scores of AGREE are much larger than that of GREE and are closer to the target value of 1. While for negative venues, the prediction scores of AGREE are closer to the target value of 0 than that of GREE. As GREE assigns the same weight for all members in the group, the model's representation ability is limited. By augmenting GREE with a learnable attention network, AGREE is capable of assigning higher weights for influential users and thus leads to better recommendation performance.

8. User #805 has traveled #58 Brazil, #31 Trukey, and #136 Japan; user #806 has traveled #547 Jiangxi, #62 Yunnan, and #553 Hunan; user #807 has traveled #139 Los Angeles, and #86 New York.

TABLE 2: Case studies of a sampled user on the effect of user-level attention (Section 4.2). The followee weights and prediction scores of the user for positive venues (Venue #23, #231, #579) and negative venues (Venue #39, #87, #357) are shown (Section 4.2).

	Model	User #144	User #425	User #696	$\hat{x}$
Venue #23	SoGREE	0.333	0.333	0.333	0.213
	SoAGREE	**0.488**	0.293	0.219	**0.481**
Venue #231	SoGREE	0.333	0.333	0.333	0.331
	SoAGREE	0.304	0.254	**0.442**	**0.610**
Venue #579	SoGREE	0.333	0.333	0.333	0.331
	SoAGREE	0.301	0.229	**0.470**	**0.638**
Venue #39	SoGREE	0.333	0.333	0.333	0.137
	SoAGREE	0.280	**0.389**	0.331	**0.037**
Venue #87	SoGREE	0.333	0.333	0.333	0.142
	SoAGREE	0.307	0.282	**0.411**	**0.086**
Venue #357	SoGREE	0.333	0.333	0.333	0.109
	SoAGREE	**0.537**	0.223	0.240	**0.053**

To gain a deep insight into the user-level attention weights learning, we performed case studies on a randomly selected testing user and investigated the attention weights for her followees (#144, #425, and #696) with respect to both positive venues (#23 Macao, #231 Shanghai, and #579 Liaoning) and negative venues (#39 Orlando, #87 Inner Mongolia, and #357 Norway). Each followee has her owning traveling history[9]. Experimental results are revealed in Table 2. We have the following observations: 1) The attention weights for followees with respect to different venues are varied. For instance, the attention weights for followee #696 are relatively high with respect to venues #231, #579, and #87. This is probably because the followee's traveled venues and the observed venues are relatively similar (they are all Chinese venues). 2) The prediction scores of SoAGREE are closer the target values (1 for positive venues and 0 for negative venues) than that of SoGREE. It illustrates the effectiveness of incorporating user-level attention network into our framework.

4.3 Overall Performance Comparison (RQ2)

Now we compare the performance of ARGEE and SoAGREE with the baselines of interest. Note that since COM,

9. Followee #144 has traveled #110 Switzerland, #130 Finland, and #566 Zurich; followee #425 has traveled #93 Miami, #215 Seattle, and #549 Chicago; followee #696 has traveled #22 Tibet, #71 Sichuan, and #864 Sinkiang.

TABLE 3: Top-N performance of both recommendation tasks for users and groups on Mafengwo (Section 4.3).

Overall Performance Comparison (Mafengwo)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
NCF	0.6363	0.5432	4.46e-06	0.4701	0.3657	1.60e-06	0.7417	0.5733	3.68e-05	0.6269	0.4141	9.20e-07
Popularity	0.4047	0.2876	2.02e-12	0.3115	0.2169	1.55e-11	0.4971	0.3172	2.09e-12	0.4251	0.2537	1.13e-11
COM	—	—	—	0.4432	0.3325	3.08e-09	—	—	—	0.5528	0.3812	2.81e-09
UL_All	—	—	—	0.4687	0.3643	8.85e-07	—	—	—	0.6252	0.4127	5.48e-07
AGR	0.6357	0.5481	6.85e-04	0.4729	0.3694	1.40e-05	0.7403	0.5738	6.26e-05	0.6321	0.4203	4.08e-05
AGREE-S	0.6369	0.5461	4.08e-05	0.4781	0.3679	5.03e-06	0.7473	0.5700	3.36e-06	0.6311	0.4186	9.68e-06
AGREE-G	0.6231	0.5377	4.19e-07	0.4661	0.3520	3.74e-08	0.7401	0.5711	6.45e-06	0.6299	0.4171	3.76e-06
NCF+avg	—	—	—	0.4774	0.3669	2.86e-06	—	—	—	0.6222	0.4140	8.84e-07
NCF+lm	—	—	—	0.4744	0.3631	5.67e-07	—	—	—	0.6302	0.4152	1.45e-06
NCF+ms	—	—	—	0.4700	0.3616	3.46e-07	—	—	—	0.6281	0.4114	3.57e-07
NCF+exp	—	—	—	0.4724	0.3647	1.03e-06	—	—	—	0.6251	0.4015	3.61e-08
AGREE	**0.6383**	**0.5502**	7.04e-07	**0.4814**	**0.3747**	8.42e-06	**0.7491**	**0.5775**	1.60e-06	**0.6400**	**0.4244**	1.04e-05
SoAGREE	**0.6510**	**0.5612**	—	**0.4898**	**0.3807**	—	**0.7610**	**0.5865**	—	**0.6481**	**0.4301**	—

TABLE 4: Top-N performance of both recommendation tasks for users and groups on CAMRa2011 (Section 4.3).

Overall Performance Comparison (CAMRa2011)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
NCF	0.6119	0.4018	1.03e-06	0.5803	0.3896	9.02e-06	0.7894	0.4535	1.89e-07	0.7693	0.4448	3.92e-07
Popularity	0.4624	0.3104	9.15e-11	0.4324	0.2825	5.92e-11	0.6026	0.3560	5.99e-11	0.5793	0.3302	3.67e-11
COM	—	—	—	0.5798	0.3785	1.20e-07	—	—	—	0.7695	0.4385	7.68e-08
UL_All	—	—	—	0.5559	0.3765	7.68e-08	—	—	—	0.7624	0.4400	1.07e-07
AGR	0.6196	0.4098	8.43e-04	0.5879	0.3933	5.62e-04	0.7897	0.4627	8.42e-06	0.7789	0.4530	2.77e-05
AGREE-S	0.6179	0.4092	2.76e-04	0.5879	0.3904	1.65e-05	0.7899	0.4632	1.21e-05	0.7739	0.4503	3.98e-06
AGREE-G	0.6125	0.4027	1.52e-06	0.5806	0.3899	1.12e-05	0.7895	0.4568	5.11e-07	0.7604	0.4455	4.92e-07
NCF+avg	—	—	—	0.5683	0.3819	2.97e-07	—	—	—	0.7641	0.4452	4.47e-07
NCF+lm	—	—	—	0.5593	0.3788	1.29e-07	—	—	—	0.7648	0.4455	4.94e-07
NCF+ms	—	—	—	0.5434	0.3710	2.75e-08	—	—	—	0.7607	0.4348	3.74e-08
NCF+exp	—	—	—	0.5648	0.3787	1.26e-07	—	—	—	0.7621	0.4426	2.05e-07
AGREE	**0.6223**	**0.4118**	—	**0.5883**	**0.3955**	—	**0.7967**	**0.4687**	—	**0.7807**	**0.4575**	—

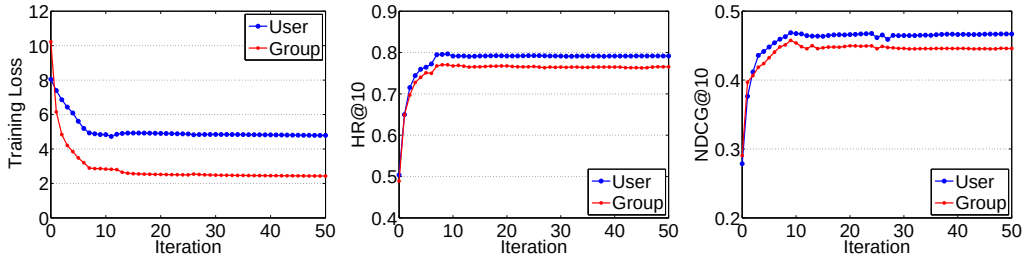
UL_All, and score aggregation methods are specially designed for group recommendation, they can not provide recommendation for individual users.

Table 3 and Table 4 show the results on Mafengwo and CAMRa2011, respectively. We have the following observations: 1) Except for SoAGREE, our AGREE method achieves the best performance on the two datasets for both recommendation tasks, significantly outperforming state-of-the-art methods (all the p-values between our model and each baseline are much smaller than 0.05, which indicates that the improvements are statistically significant). This validates the effectiveness of our AGREE solution, more specifically, the positive effect of the attention network in aggregating the preference of group members and simultaneously addressing the two tasks. 2) The performance of neural network-based solutions (i.e., NCF, AGR, NCF+avg, NCF+lm, NCF+ms, NCF+exp, AGREE, and SoAGREE) are superior to that of non-personalized approach (Popularity), probabilistic graphical model (COM), and aggregation method (UL_All). This demonstrates the superiority of neural networks, especially their great ability in modeling the high-order interactions among users, groups, and items. 3) There is no obvious winner among the score aggregation-based solutions. For example, NCF+avg outperforms NCF+lm when $K = 5$ on Mafengwo, but underperforms when $K = 10$. This again confirms that a predefined, static score aggregation strategy is insufficient to predict the group decision well. In contrast, AGREE dynamically associates weights for group members by learning from data, which shows remarkable flexibility and superiority. 4) On the Mafengwo dataset, SoAGREE outperforms AGREE by a great margin. This demonstrates the effectiveness of considering social followee information, which employs the attention mechanism as the underlying principle for aggregating individual followees. 5) AGREE beats AGREE-S by a great margin for both user-item recommendation and group-item recommendation. The user-item interactions and group-item interactions are rather sparse. Jointly optimizing user-item interactions and group-item interactions with shared hidden layers could effectively alleviate the data sparsity issue. That is why the performance of AGREE is superior to that of AGREE-S. 6) The experimental results of AGREE are superior to that of AGREE-G. Although the group member information is utilized to construct user embeddings, the performance of AGREE-G even degrades to a large extent. This is probably because the groups are temporarily organized and the relationship among group members are not very tight. Therefore, the group information is not suitable to the construction of user embeddings in our scenario.

4.4 Convergence Analysis and Parameter Tuning (RQ3)

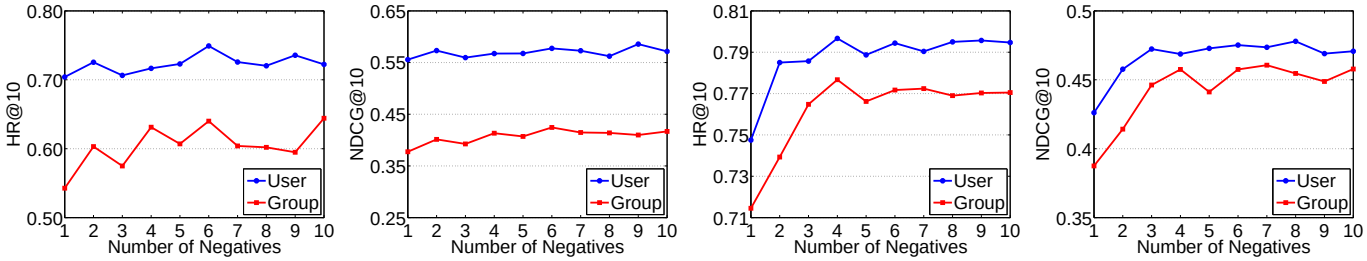
In order to demonstrate the robustness and effectiveness of our proposed framework, we investigated the convergence of AGREE and meticulously studied the sensibility of several factors, such as the number of negative samples and the dropout ratio.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2936475, IEEE Transactions on Knowledge and Data Engineering



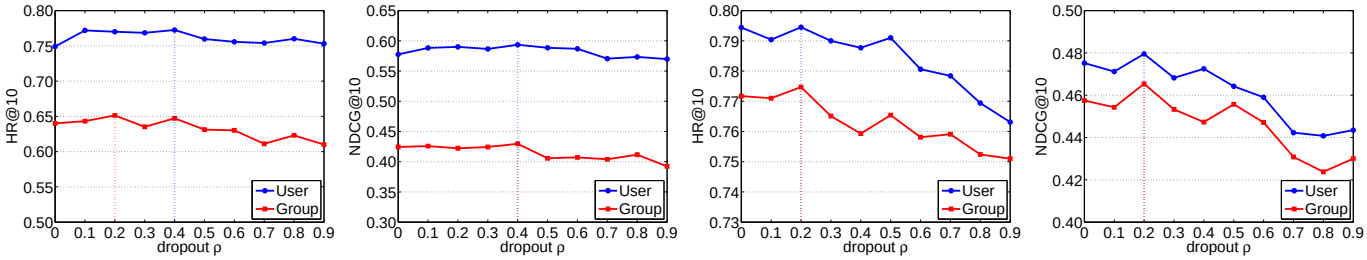
(a) Training Loss **(b) HR@10** **(c) NDCG@10**

Fig. 5: Training loss and recommendation performance of AGREE w.r.t. the number of iterations on CAMRa2011.



(a) Mafengwo — HR@10 **(b) Mafengwo — NDCG@10** **(c) CAMRa2011 — HR@10** **(d) CAMRa2011 — NDCG@10**

Fig. 6: Performance of AGREE w.r.t. the number of negative samples for each positive instance.



(a) Mafengwo — HR@10 **(b) Mafengwo — NDCG@10** **(c) CAMRa2011 — HR@10** **(d) CAMRa2011 — NDCG@10**

Fig. 7: Performance of AGREE w.r.t. the dropout ratio ρ.

Convergence: We recorded the value of training loss, HR@10, and NDCG@10 along with each iteration using the optimal parameter setting. Figure 5a, 5b, and 5c show the training loss, HR@10, and NDCG@10 with the increasing number of iterations on the CAMRa2011 dataset. The convergence results for the Mafengwo dataset is almost the same as that of CAMRa2011 dataset. To save the space, we only illustrated the convergence results of the CAMRa2011 dataset here. We have the following observations: 1) With more iterations, the training loss of AGREE gradually decreases and the recommendation performance is improved. AGREE converges fast in the first 10 iterations, and reaches its optimal results around the 20th iteration. This indicates the rationality of our learning scheme. 2) Jointly observing Figure 5b and 5c, the performance of NDCG@10 fluctuates markedly over the iterations, while the performance of HR@10 is relatively stable. It is reasonable since NDCG@10 not only measures whether the test item is presented on the top-10 list, but also accounts for the position of the hit (which is ignored in HR@10).

Impact of Negative Samples: The strategy of negative sampling has been proven rational and effect in [7], [14]. It randomly samples various numbers of missing data as negative samples to pair with each positive instance. With more negative samples selected, the performance of negative sampling becomes stable and approximates the result of all missing data considered. To illustrate the impact of negative sampling for AGREE, we show the performance of AGREE w.r.t. different negative sample ratios on both Mafengwo and CAMRa2011 datasets in Figure 6. We have the following observations: 1) It is obviously seen that one negative sample for each positive instance is not optimal for the final performance, and sampling more negative samples is beneficial. Compared with traditional pairwise sampling method which selects only one negative sample to pair with each positive sample, such as BPR [42], AGREE shows the advantage of selecting flexible sampling ratio for negative instances. 2) With more negative samples selected, the performance of AGREE becomes stable and reaches its optimal results. For both datasets, the optimal sampling ratio is around 4 to 6, and that is why the number of negative samples is set to 4 as illustrated in Section 4.1.1.

Impact of Dropout: Although deep neural networks have the great ability in representation learning, a deep architecture easily leads to the overfitting issue due to the limited training data. To prevent AGREE and SoAGREE from overfitting, we employed the dropout strategy to improve the regularization of our deep model. In particular, we randomly dropped ρ of neurons on pooling layers and hidden layers, whereinto ρ is the dropout ratio. Figure 7 reveals the performance of AGREE w.r.t. the dropout ratio ρ on both datasets (SoAGREE reveals similar results and are omitted here). We have the following observations: 1) When the dropout ratio equals to 0, AGREE performs poor which

TABLE 5: Top-N performance of AGREE and its two simplified variants on the Mafengwo dataset (Section 4.5).

Component Performance Comparison (Mafengwo)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
AGREE-UE	0.6220	0.5364	2.80e-07	0.4141	0.3322	2.99e-09	0.7309	0.5716	9.02e-06	0.5709	0.3832	3.39e-09
AGREE-GE	0.6363	0.5432	4.46e-06	0.4291	0.3405	7.18e-09	0.7417	0.5733	3.68e-05	0.6181	0.4020	3.95e-08
AGREE	**0.6383**	**0.5502**	—	**0.4814**	**0.3747**	—	**0.7491**	**0.5775**	—	**0.6400**	**0.4244**	—

TABLE 6: Top-N performance of AGREE and its two simplified variants on the CAMRa2011 dataset (Section 4.5).

Component Performance Comparison (CAMRa2011)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
AGREE-UE	0.6043	0.3945	1.12e-07	0.5793	0.3832	4.47e-07	0.7601	0.4465	4.09e-08	0.7441	0.4376	6.36e-08
AGREE-GE	0.6119	0.4018	1.03e-06	0.5803	0.3896	9.02e-06	0.7894	0.4535	1.89e-07	0.7593	0.4448	3.92e-07
AGREE	**0.6223**	**0.4118**	—	**0.5883**	**0.3955**	—	**0.7967**	**0.4687**	—	**0.7807**	**0.4575**	—

TABLE 7: Top-N performance of SoAGREE and its two simplified variants on the Mafengwo dataset (Section 4.5).

Component Performance Comparison (Mafengwo)												
	K=5						K=10					
	User			Group			User			Group		
	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value	HR	NDCG	p-value
SoAGREE-FE	0.6305	0.5419	7.21e-08	0.4728	0.3680	3.93e-07	0.7406	0.5713	1.89e-07	0.6311	0.4187	6.09e-07
SoAGREE-UE	0.6383	0.5502	7.04e-07	0.4814	0.3747	8.42e-06	0.7491	0.5775	1.60e-06	0.6400	0.4244	1.04e-05
SoAGREE	**0.6510**	**0.5612**	—	**0.4898**	**0.3807**	—	**0.7610**	**0.5865**	—	**0.6481**	**0.4301**	—

is caused by the overfitting. 2) The optimal settings for dropout ratio locate on 0.2 to 0.4 on both datasets. When the dropout ratio exceeds the optimal settings, the performance of AGREE decreases dramatically, which suffers from the insufficient information.

4.5 Importance of Components (RQ4)

The overall performance comparison shows that AGREE and SoAGREE obtain the best results, demonstrating the effectiveness of the integrated end-to-end solution. To further understand the importance of components in attentive user representation learning and attentive group representation learning, we performed some ablation studies. For convenience, we use the name AGREE-UE to denote the method "AGREE with user embedding aggregation only", AGREE-GE to denote "AGREE with group preference embedding only" (which is equivalent to the NCF method), SoAGREE-FE to denote the method "SoAGREE with followee embedding aggregation only", and SoAGREE-UE to denote "SoAGREE with user preference embedding only" (which is equivalent to the AGREE method). It is worth noting that the performance of SoAGREE is only evaluated on the Mafengwo dataset. Therefore, the component performance comparison results of SoAGREE-FE, SoAGREE-UE, and SoAGREE are just evaluated on Mafengwo as well.

Table 5, Table 6, and Table 7 show the results of AGREE, SoAGREE and their corresponding simplified variants. We have the following observations: 1) AGREE consistently and significantly outperforms AGREE-UE and AGREE-GE on both datasets with respect to both metrics, which can be evidenced by the small p-values. This indicates that both components of user embedding aggregation and group preference embedding are beneficial to model group decisions, and combining them leads to better performance. Similar component performance comparison results are also revealed on SoAGREE, SoAGREE-FE, and SoAGREE-UE on the Mafengwo dataset. 2) AGREE-GE shows better performance than AGREE-UE on both datasets. This reveals that the group preference embedding has a larger impact in learning group representation in our method. 3) The performance of SoAGREE-UE is superior to that of SoAGREE-FE, which indicates that the importance of the user preference embedding learning is superior to that of the followee embedding aggregation learning on the SoAGREE framework.

5 Conclusion and Future Work

In this work, we address the group recommendation problem from the perspective of neural representation learning. Under the framework, there are two key factors to estimate a group's preference on an item well: 1) how to obtain a semantic representation for a group, and 2) how to model the interaction between a group and an item. We propose a novel solution named AGREE, which addresses the first factor of group representation learning with an attention network and the second factor of interaction learning with NCF. Specifically, by leveraging the attention network, AGREE can automatically learn the importance of a group member from data; by leveraging NCF, it is capable of learning the complicated interactions among groups, users, and items. Moreover, social followee information is further incorporated into the framework, and is termed as SoAGREE. In SoAGREE, the followees are regarded as attributes of a user and are aggregated via another attention network to dynamically adjust the attention weights for followees. Thereafter, we integrate the modeling of user-item interaction data into AGREE, allowing the two tasks of recommending items for groups and users to be mutually reinforced. To validate the effectiveness of AGREE and SoAGREE, we perform extensive experiments on two real-world datasets. The results show that AGREE and

SoAGREE achieve state-of-the-art performance for group recommendation; further micro-level analyses demonstrate how the attention networks improve the performance, how predefined hyper-parameters affect our methods, and how components of AGREE and SoAGREE affect the results.

In future, we plan to extend our work in the following two directions. First, we are interested in realizing group recommender systems in an online fashion. The interests of users evolve over time, and so do the preferences of groups. As it is computationally prohibitive to retrain a recommender model in real-time, it would be extremely helpful to do online learning. Along this line, we are particularly interested in leveraging reinforcement learning methods to provide online recommendation. Second, we will study how to alleviate the troublesome cold-start and data sparsity problems in recommendation by utilizing multimedia objects (e.g., reviews, images, and videos). The maturing of multimedia techniques in recent years provides us a great opportunity to inject multimedia content into recommendation, which we believe is a promising direction in improving the performance of recommendation.

ACKNOWLEDGMENTS

This work is supported by the Fundamental Research Funds for the Central Universities, the National Science Foundation for Young Scientists of China (No. 6180070114 and No. 61702173), the National Science Foundation of China (No. 61772190), the National Key Research and Development Program of China (No. 2018YFB07040 and No. SQ2018YFB140002), the Natural Science Foundation of Hunan Province (NO. 2019JJ50057) and the Hunan Key Research and Development Program of China (No. 2016JC2015). This work is also part of NExT research, supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

REFERENCES

[1] Y. Ren, M. Tomko, F. D. Salim, J. Chan, C. L. Clarke, and M. Sanderson, "A location-query-browse graph for contextual recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 204–218, 2018.

[2] X. He, M. Gao, M.-Y. Kan, and D. Wang, "Birank: Towards ranking on bipartite graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 57–71, 2017.

[3] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4649–4661, 2012.

[4] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, "Learning on big graph: Label inference and regularization with anchor hierarchy," *IEEE transactions on knowledge and data engineering*, vol. 29, no. 5, pp. 1101–1114, 2017.

[5] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1864–1877, 2016.

[6] D. Cao, L. Nie, X. He, X. Wei, J. Shen, S. Wu, and T.-S. Chua, "Version-sensitive mobile app recommendation," *Information Sciences*, vol. 381, pp. 161–175, 2016.

[7] D. Cao, X. He, L. Nie, X. Wei, X. Hu, S. Wu, and T.-S. Chua, "Cross-platform app recommendation by jointly modeling ratings and texts," *ACM Transactions on Information Systems*, vol. 35, no. 4, p. 37, 2017.

[8] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proceedings of the ACM Conference on Recommender Systems*. ACM, 2010, pp. 119–126.

[9] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 754–765, 2009.

[10] L. Boratto and S. Carta, "State-of-the-art in group recommendation and new approaches for automatic identification of groups," in *Information Retrieval and Mining in Distributed Environments*. Springer, 2010, pp. 1–20.

[11] E. Quintarelli, E. Rabosio, and L. Tanca, "Recommending new items to ephemeral groups using contextual user influence," in *Proceedings of the ACM Conference on Recommender Systems*. ACM, 2016, pp. 285–292.

[12] J. Chen, H. Zhang, X. He, W. Liu, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 335–344.

[13] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proceedings of International Joint Conference on Artificial Intelligence*. AAAI, 2017, pp. 3119–3125.

[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the International Conference on World Wide Web*. ACM, 2017, pp. 173–182.

[15] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2018.

[16] Y.-D. Seo, Y.-G. Kim, E. Lee, K.-S. Seol, and D.-K. Baik, "An enhanced aggregation method considering deviations for a group recommendation," *Expert Systems with Applications*, vol. 93, pp. 299–312, 2018.

[17] Q. Yuan, G. Cong, and C.-Y. Lin, "Com: a generative model for group recommendation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 163–172.

[18] X. Liu, Y. Tian, M. Ye, and W.-C. Lee, "Exploring personal impact for group recommendation," in *Proceedings of the ACM International Conference on Information and Knowledge Management*. ACM, 2012, pp. 674–683.

[19] T. D. Q. Vinh, T. A. N. Pham, G. Cong, and X. L. Li, "Attention-based group recommendation," *arXiv preprint arXiv:1804.04327*, 2018.

[20] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao, "Deep modeling of group preferences for group-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2014, pp. 1861–1867.

[21] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *Proceedings of the International Conference on Data Engineering*. IEEE, 2019, pp. 566–577.

[22] L. Sun, X. Wang, Z. Wang, H. Zhao, and W. Zhu, "Social-aware video recommendation for online social groups," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 609–618, 2017.

[23] J. Gorla, N. Lathia, S. Robertson, and J. Wang, "Probabilistic group recommendation via information matching," in *Proceedings of the International Conference on World Wide Web*. ACM, 2013, pp. 495–504.

[24] M. Wang, Y. Gao, K. Lu, and Y. Rui, "View-based discriminative probabilistic modeling for 3d object retrieval and recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1395–1407, 2013.

[25] X. Meng, S. Wang, K. Shu, J. Li, B. Chen, H. Liu, and Y. Zhang, "Personalized privacy-preserving social recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2018.

[26] J. Zhang, Y. Yang, Q. Tian, L. Zhuo, and X. Liu, "Personalized social image recommendation method based on user-image-tag model," *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2439–2449, 2017.

[27] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, and L. Guo, "Social recommendation with an essential preference space," in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2018.

[28] L. Gao, J. Wu, Z. Qiao, C. Zhou, H. Yang, and Y. Hu, "Collaborative social group influence for event recommendation," in *Proceedings of the ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1941–1944.

[29] A. Salehi-Abari and C. Boutilier, "Preference-oriented social networks: Group recommendation and inference," in *Proceedings of the ACM Conference on Recommender Systems*. ACM, 2015, pp. 35–42.
[30] Y. Liu, C. Liang, F. Chiclana, and J. Wu, "A trust induced recommendation mechanism for reaching consensus in group decision making," *Knowledge-Based Systems*, vol. 119, pp. 221–231, 2017.
[31] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 355–364.
[32] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 185–194.
[33] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 5659–5667.
[34] Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An, "Enhancing recurrent neural networks with positional attention for question answering," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 993–996.
[35] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, and M. Wang, "Low-rank multi-view embedding learning for micro-video popularity prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2017.
[36] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: Neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
[37] P. Sun, L. Wu, and M. Wang, "Attentive recurrent social recommendation," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2018, pp. 185–194.
[38] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 235–244.
[39] L. Liao, X. He, H. Zhang, and T. S. Chua, "Attributed social network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2017.
[40] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, 2015, pp. 1–15.
[41] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, and Z. Qin, "Aesthetic-based clothing recommendation," in *Proceedings of the International Conference on World Wide Web*. ACM, 2018, pp. 649–658.
[42] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 452–461.
[43] D. Cao, L. Nie, X. He, X. Wei, S. Zhu, and T.-S. Chua, "Embedding factorization models for jointly recommending items and user generated lists," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 585–594.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.
[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[46] S. Seko, T. Yagi, M. Motegi, and S. Muto, "Group recommendation using feature space representing behavioral tendency and power balance among members," in *Proceedings of the ACM Conference on Recommender Systems*. ACM, 2011, pp. 101–108.
[47] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the ACM Conference on Recommender Systems*. ACM, 2010, pp. 39–46.
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

Da Cao received the Master degree and Ph.D. degree from Xiamen University in 2013 and 2017, respectively. He is currently an assistant professor in the College of Computer Science and Electronic Engineering, Hunan University. His research interests include recommender systems, multimedia information retrieval, and natural language processing. Various parts of his work have been published in top forums including SIGIR, TOIS, and INS. He has been served as reviewers for various journals and conferences including SIGIR, SIGKDD, TKDE, TKDD, INS, and KBS.

Xiangnan He received the Ph.D. degree in Computer Science from National University of Singapore in 2016. He is currently a professor in the University of Science and Technology of China. His research interests span information retrieval, data mining, and multimedia analytics. He has over 60 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE, TOIS, and TMM. His work on recommender systems has received the Best Paper Award Honourable Mention in WWW 2018 and ACM SIGIR 2016. Moreover, he has served as the PC chair of CCIS 2019, area chair of MM 2019 and CIKM 2019, and PC member for several top conferences including SIGIR, WWW, KDD etc., and the regular reviewer for journals including TKDE, TOIS, TMM, etc.

Lianhai Miao received the BEc. degree in the School of Science from Nanchang University in 2017. Currently, he is working toward the Master degree in the College of Computer Science and Electronic Engineering, Hunan University. He works in the fields of recommender systems, multimedia techniques, and machine learning. His current research interests focus on explainable recommendation and recommendation technologies for multimedia content. He is a student member of CCF and ACM.

Guangyi Xiao received the BEc. degree in Software Engineering from Hunan University in 2006, the Master degree and Ph.D. degree in Software Engineering from University of Macau in 2009 and 2015, respectively. He is currently an assistant professor in the College of Computer Science and Electronic Engineering, Hunan University. His research interests include natural language processing, data mining, and recommender systems. His work has appeared in international journals and conferences.

Hao Chen received the Ph.D degree from Central South University in 2012. He is currently a professor and doctoral tutor in the College of Computer Science and Electronic Engineering, Hunan University. His main research interests include recommender systems, data mining, and multimedia information retrieval. He has several work appeared on top venues and has been served as reviews for various journals and conferences.

Jiao Xu received the Master degree in Software Engineering from Central South University in 2015. She is currently a researcher in in the Central Research Institute, CVTE Inc. Her research interests span recommender systems, computational advertising, multimedia mining, and knowledge graph. She currently works in the fields of multimodal data fusion and recommendation technologies for multimedia content. Her work has appeared in several international journals and conferences.